

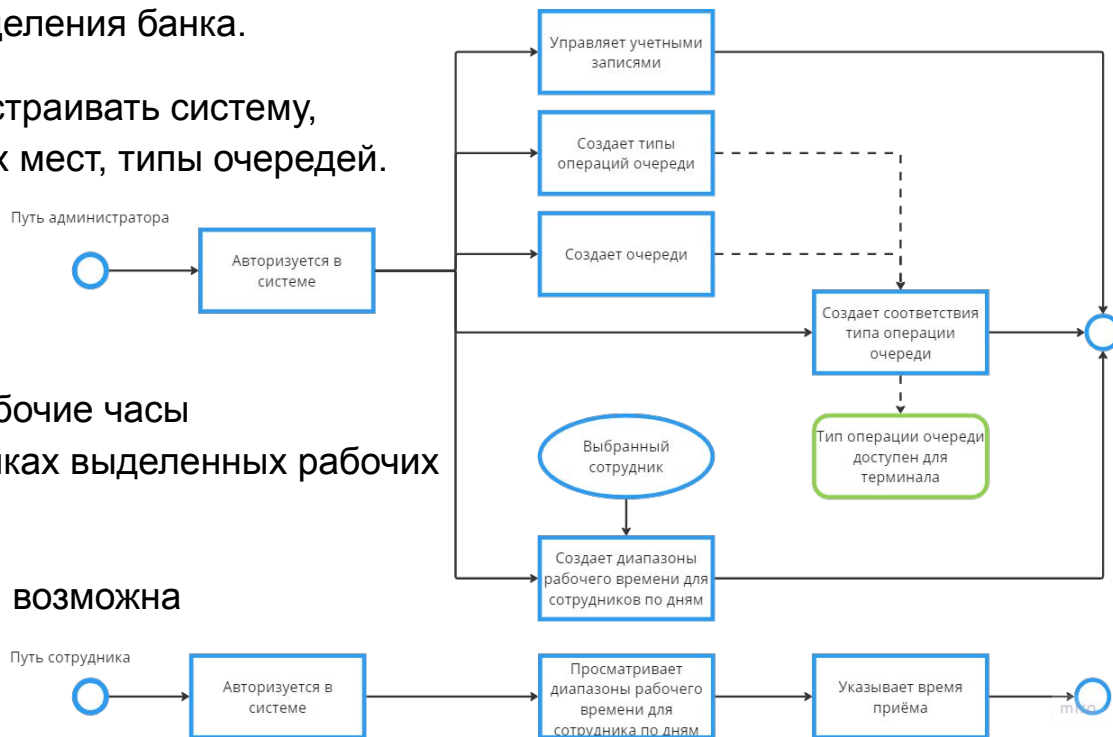
# Электронная очередь: функциональная структура

Система электронной очереди функционирует автономно в рамках одного отделения банка.

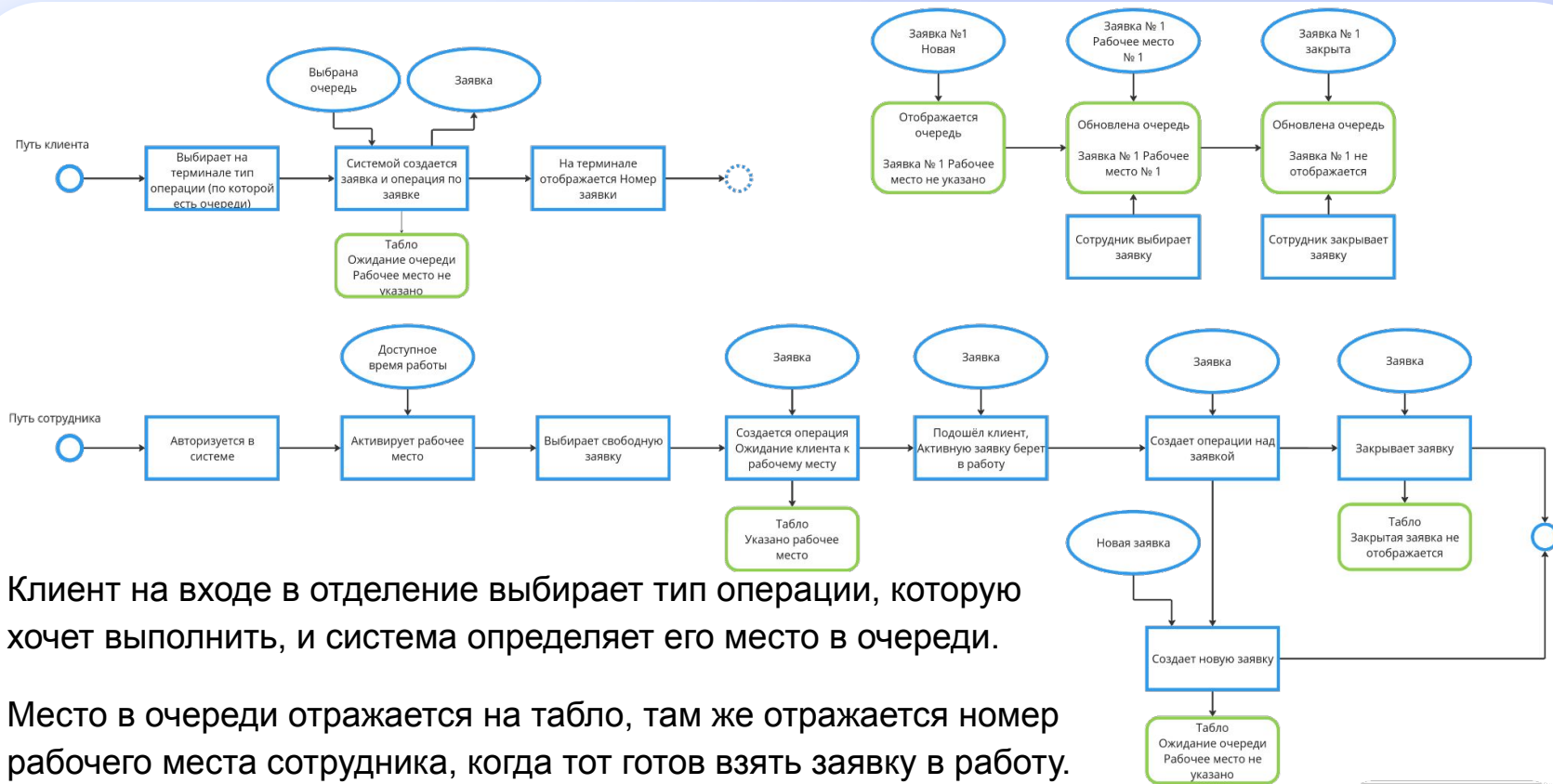
Позволяет администратору настраивать систему, установить количество рабочих мест, типы очередей.

Администратор определяет рабочие часы сотрудников. Сотрудники в рамках выделенных рабочих часов указывают часы приёма.

Обработка заявок сотрудником возможна только в его часы приёма.



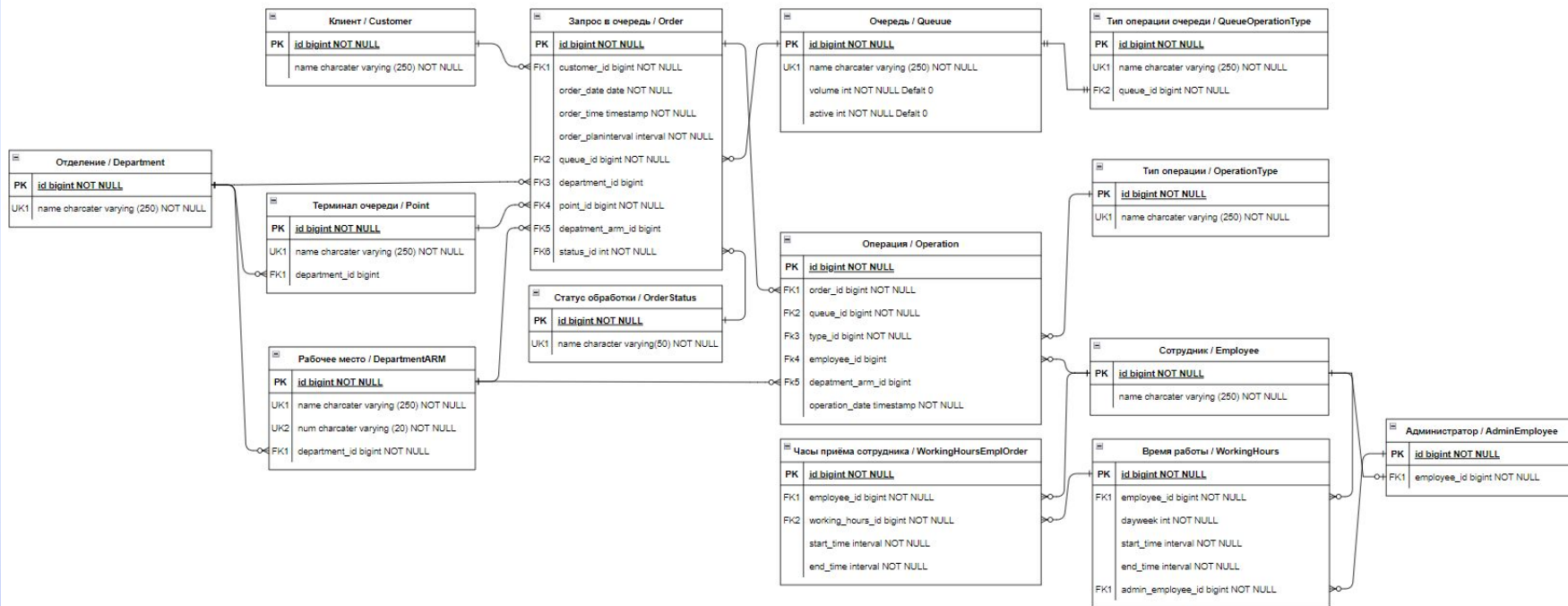
# Электронная очередь: функциональная структура



Клиент на входе в отделение выбирает тип операции, которую хочет выполнить, и система определяет его место в очереди.

Место в очереди отражается на табло, там же отражается номер рабочего места сотрудника, когда тот готов взять заявку в работу.

# Электронная очередь: архитектура данных



# Электронная очередь: архитектура системы

В постановке нет требований о централизованном управлении и/или мониторинге отделений.

При централизации системы в банке есть риски скорости и доступности сетевого соединения.

При локализации системы в отделении требуется организовать развертывание и эксплуатацию.

В случае локализации системы в отделении, при отключении электричества возможен выход из строя локального компьютера, на котором хранится состояние очередей.

Для резервирования состояния потребуется несколько компьютеров, объединённых репликацией. Это могут быть выделенные для системы машины, но сколько?!

В отделении уже находятся компьютеры сотрудников, администратора, клиентский терминал и информационное табло. Что, если использовать их для репликации?

В таком случае потребуется создать одноранговую сеть, в которой будет реализовано приложение с логикой ведения очередей. Такая сеть – **распределенный реестр**.

# Электронная очередь: архитектура системы



# Электронная очередь: API модуля расписаний

Модуль расписаний представляет собой смарт-контракт на Solidity.

Он вызывается из фронтенда приложения администратора или приложения сотрудника через JSON RPC API узла сети Quorum, запущенного на данном рабочем месте.

Для вызова методов, изменяющих состояние смарт-контракта расписаний требуется электронная подпись, созданная при помощи ключа сотрудника или администратора, ID которого заранее добавлен в ролевой модели смарт-контракта.

```
interface IEmployeeSchedule {  
    // Установка рабочих часов для сотрудника  
    function setEmployeeWorkingHours(address employee, uint startTime, uint endTime) external;  
    // Установка часов приёма для сотрудника  
    function setReceptionHours(uint startTime, uint endTime) external;  
    // Получение рабочих часов сотрудника  
    function getEmployeeWorkingHours(address employee) external view returns (uint, uint);  
    // Получение часов приёма сотрудника  
    function getEmployeeReceptionHours(address employee) external view returns (uint, uint);  
    // Проверка, находится ли текущее время в рамках часов приёма сотрудника  
    function isDuringReceptionHours(address employee) external view returns (bool);  
}
```

# Электронная очередь: API модуля очередей

Модуль очередей также представляет собой смарт-контракт, учитывающий типы операций, их отображение на доступные в отделении очереди, а также логику добавления заявки в очередь с клиентского терминала и её взятия в работу и закрытия сотрудником.

```
interface IOperationsManagement {  
    // Методы для администратора  
    // Инициализация контракта с адресом контракта расписаний  
    function initialize(address scheduleContract) external;  
    // Добавление нового типа операции  
    function addOperationType(string memory operationType) external;  
    // Удаление существующего типа операции  
    function removeOperationType(string memory operationType) external;  
    // Добавление новой очереди  
    function addQueue(uint queueId) external;  
    // Удаление существующей очереди  
    function removeQueue(uint queueId) external;  
    // Связывание типа операции с очередью  
    function linkOperationToQueue(string memory operationType, uint queueId) external;  
    // Отвязывание типа операции от очереди  
    function unlinkOperationFromQueue(string memory operationType) external;
```



# Электронная очередь: API модуля очередей

```
function linkOperationToQueue(string memory operationType, uint queueId) external;
// Отвязывание типа операции от очереди

function unlinkOperationFromQueue(string memory operationType) external;

// Методы для клиентского терминала
// Создание новой заявки и возвращение её ID
function createRequest(string memory operationType) external returns (uint);

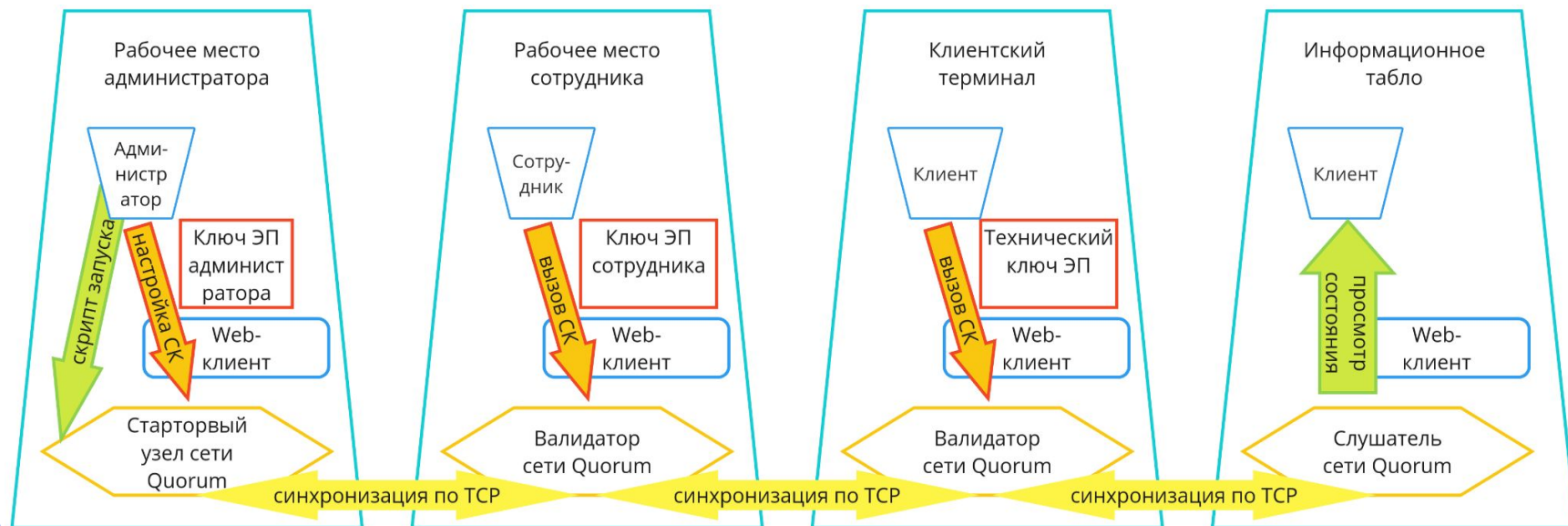
// Методы для сотрудника
// Регистрация рабочего места для сотрудника
function registerWorkstation(uint workstationId) external;
// Взятие следующей заявки из указанной очереди
function takeNextRequest(uint queueId) external returns (uint);
// Закрытие указанной заявки
function closeRequest(uint requestId) external;

// Методы для информационного табло
// Получение всех открытых заявок
function getOpenRequests() external view returns (uint[] memory);
// Получение всех заявок в работе с их идентификаторами рабочих мест
function getWorkInProgressRequests() external view returns (uint[] memory, uint[] memory);
```



# Электронная очередь: архитектура развертывания

Развертывание системы может выполняться на компьютерах рабочих мест, клиентских терминалах и информационных табло отделения согласно политикам развертывания банка в виде дистрибутивов сети распределенного реестра Quorum и Web-клиента, доступного в виде статических файлов для просмотра в системном браузере.



# Электронная очередь: функциональные требования

## Клиент

1. Клиент имеет возможность просмотреть список операций, которые могут быть выполнены в отделении (*конкретная очередь 1-ко-многим операциям, операция 1-к-1 к конкретной очереди или одна операция соответствует конкретной очереди, в одну очередь могут попасть несколько операций*).
2. Клиент имеет возможность инициировать запрос на постановку в очередь для обслуживания выбрав тип операции и нажав кнопку «в очередь» (можно сделать без кнопки «в очередь» - просто «пиктограммы» или список операций и есть кнопка – нажал, вернулся номер в очереди). Расширения – система присваивает следующий номер заявке в очереди/этот номер отображается на экране клиентского терминала/этот номер отображается на экране информационного табло.
3. Клиент имеет возможность вернуться на экран просмотра списка операций из операции или из отображаемого номера очереди.

# Электронная очередь: функциональные требования

## Сотрудник

4. Сотрудник отделения имеет возможность зарегистрироваться в системе на определенном рабочем месте. Расширение – отказ в регистрации если зарегистрированных сотрудников = количеству рабочих мест, настроенному администратором

Для зарегистрированных сотрудников отделения:

5. Сотрудник отделения имеет возможность просмотреть все текущие запросы всех очередей

6. Сотрудник отделения имеет возможность инициировать обработку заявки из выбранной очереди (не определенную заявку – первую из выбранной очереди). Расширение – система выдает ему номер заявки из выбранной очереди с учетом конкуренции сотрудников отделения (описано ранее), на табло автоматически отображается номер заявки и номер рабочего места этого сотрудника отделения /либо отказ (нет заявки в очереди).

7. Сотрудник отделения имеет возможность инициировать завершение обработки ранее полученной заявки. (расширение успешной ветки предыдущего US).

8. Сотрудник отделения имеет возможность просмотреть свои рабочие часы (ограничение - не после инициации обработки заявки).

9. Сотрудник отделения имеет возможность изменить свои рабочие часы (ограничение - не после инициации обработки заявки).

10. Сотрудник отделения имеет возможность выйти из системы (прекратить регистрацию на рабочем месте).

# Электронная очередь: функциональные требования

## Администратор

11. Администратор имеет возможность зарегистрироваться в системе.

Зарегистрированный администратор:

12. Администратор имеет возможность управлять учетными записями (логины/пароли/права) сотрудников отделения.

13. Администратор имеет возможность вести (добавлять, редактировать, удалять) список типов операций в отделении.

14. Администратор имеет возможность вести список очередей (с указанием правила формирования номера в очереди)

15. Администратор имеет возможность ведения соответствия типа операции-очереди.

16. Администратор имеет возможность просмотреть и изменить диапазоны рабочего времени для всех или выбранных сотрудников отделения по дням недели.

17. Система имеет возможность вернуть новый номер из конкретной очереди для нового запроса на постановку в очередь;

18. Система имеет возможность вернуть первый номера заявки из конкретной очереди при инициации обработки заявки из выбранной очереди сотрудником отделения;

19. Система имеет возможность разрешить коллизию получения номера заявки из очереди при одновременной инициации запроса на постановку в очередь для обслуживания для нескольких клиентов

20. Система имеет возможность разрешить коллизию получения номера заявки из очереди при одновременной инициации обработки заявки из выбранной очереди несколькими сотрудниками отделения

21. Система имеет возможность проверки количества зарегистрированных сотрудников отделения при регистрации очередного сотрудника.

22. Система имеет возможность проверки попадания текущего времени в диапазоны рабочего времени и часов приема для сотрудника отделения при инициации обработки заявки.

23. Система имеет возможность очистки очереди при завершении дня (закрытие незакрытых инициализированных запросов/удаление не обработанных запросов)